SAND REPORT
SAND2003-3609
Unlimited Release
Printed October 2003

The Sandia Petaflops Planner

Erik P. DeBenedictis

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/ordering.htm

SAND2003-3609
Unlimited Release
Printed October 2003

The Sandia Petaflops Planner

Erik P. DeBenedictis
Scalable Computing Systems
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-1110

Abstract

The Sandia Petaflops Planner is a tool for projecting the design and performance of parallel supercomputers into the future. The mathematical basis of these projections is the International Technology Roadmap for Semiconductors (ITRS, or a detailed version of Moore's Law) and DOE balance factors for supercomputer procurements. The planner is capable of various forms of scenario analysis, cost estimation, and technology analysis. The tool is described along with technology conclusions regarding PFLOPS-level supercomputers in the upcoming decade.

Intentionally Left Blank

Contents

Figures

Tables

Nomenclature

ASCI…… Advanced Simulation and Computing (the “I” is silent)
ASIC……Application Specific Integrated Circuit
DOE….. .....Department of Energy
DRAM…..Dynamic Random Access Memory
FLOPS…….. .....Floating Operations Per Second
GFLOPS…… 10^9 Floating Operations per Second
IPC IO….. .....Interprocessor Communications Input Output (I. e. MPI)
ITRS…..... International Technology Roadmap for Semiconductors
MPI……Message Passing Interface
MPP……. .....Massively Parallel Processor
MPU……. .....MicroProcessor Unit
PetaFLOPS……. ..... 10^{15} Floating Operations Per Second
PIM…….....Processor In Memory
SNL….. .....Sandia National Laboratories
TFLOPS….. ..... 10^{12} Floating Operations Per Second

Intentionally Left Blank

Introduction

Over the past 20 years, parallel supercomputers have gone from an academic curiosity to vital tools for science, defense, engineering, and a host of other fields. The Department of Energy's (DOE's) Advanced Simulation and Computing (ASCI) program has driven the development of parallel supercomputers through a staged series of procurements up to the current programmatic limit of 100 TFLOPS. It seems evident that applications will continue to want more power, causing us to wonder if the computer technology is up to the task?

To answer this, we need to find an oracle to foretell the future of supercomputers. We found our oracle in Moore's Law, or more specifically the extensive body of knowledge known as the ITRS roadmap developed by the semiconductor industry projecting future progress in integrated circuits. To apply Moore's Law for integrated circuits to supercomputers, we need an abstracted definition of a supercomputer that can be applied to hypothetical supercomputers in the future. We found this definition in the procurement rules for DOE supercomputers. Over the half-dozen generations of DOE supercomputer procurements, DOE labs have figured out how to specify the relationships between a supercomputer's computing rate, memory size, communications rate, etc. needed for a machine to work well for applications. Adding Moore's Law to DOE's procurement rules yields hundreds of parameters and mathematical relationships – but which can be solved with effort.

We wrote a "Petaflops Planner" program that essentially projects popular supercomputer designs into the future using Moore's Law. We wrote the planner as a tool for multiple purposes: testing hypotheses on computer architecture, optimizing designs, projecting costs, etc.

We then used the planner to see if computer technology was up to the task of PFLOPS-level supercomputers. The body of the paper explains how the answer can be "yes," but only by employing new technology.

Previous Work

Balance factors

Sandia and other DOE laboratories have had criteria for specifying supercomputers in procurements. These criteria have been applied to supercomputers with vastly different performance levels over the last 20 years. As a result, the criteria have become scalable rules that define supercomputers that will perform well within the DOE.

	DATA-INTENSIVE (RED STORM[1])	COMPUTE-INTENSIVE (PURPLE[2])
Memory Bytes/FLOP	1.0 TFLOPS$^{-.25}$	1.0 TFLOPS$^{-.25}$
Memory Bandwidth Bytes/FLOP	4	1
Peak IPC IO Bytes/sec/FLOP	2	.1/12.≈.01
Total IPC IO Bytes/sec/FLOP	2	.1
Network bisection bandwidth Bytes/sec/FLOP	.075	.05

Table 1: Balance Factors

These scalable rules are based on "balance factors"[ref 1, 2]. A DOE laboratory begins by specifying a performance target – originally measured as peak GFLOPS, now peak TFLOPS. The balance factors specify the values of other parameters as a function of the peak FLOPS. DOE procurements tend to use similar balance factors, but with different values representing a different application mix (table 1).

Architectures and Packaging Styles

The supercomputing community has just a few leading architectures or design styles:

- Virtually all supercomputer today are clusters of Symmetric MultiProcessors (SMPs, figure 1), which are composed of separate commodity microprocessor chips, memory chips, and a router implemented as an Application Specific Integrated Circuit (ASIC).

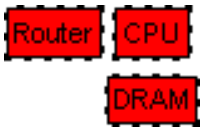

Figure 1: An SMP Node

- nCUBE Corporation pioneered a different implementation style where the CPU and router are both in the same ASIC chip, but with separate memory (figure 2). IBM is using this implementation style for the DOE Blue Light project.

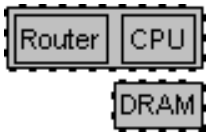

Figure 2: An nCUBE Node

- Processor-In-Memory (PIM) represents a more radical departure. PIMs include processors, memory, and router in a single chip – often hundreds of each (figure 3). PIM systems may consist of homogeneous arrays of PIM chips, but may also include additional conventional memory. There are different views on the best "instruction set" for PIMs, and some designs may represent a different architecture as well as a different packaging style.

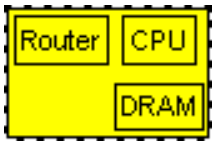

Figure 3: A PIM Node

Semiconductor Roadmap

The ITRS roadmap[ref. 3] is a project by the semiconductor industry to track the progress of semiconductor technology, identifying roadblocks before they become problems. While Moore's Law is a one-sentence statement, the ITRS Roadmap is a time comparable in size to a telephone book (figure 4).

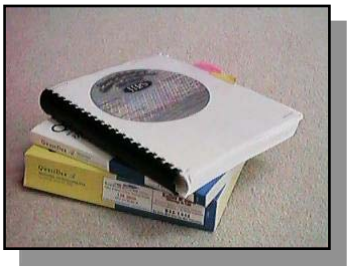

Figure 4: The ITRS compared to the Albuquerque Telephone Book

A sample from the 1999 ITRS roadmap illustrates the process. The Semiconductor Industries Association (SIA) establishes a panel of experts in each of about a dozen aspects of semiconductor technology (process technology, packaging, DRAM design, testing, …). These experts analyze progress in their area, either confirming that Moore's Law can be maintained or identifying obstacles. To fit on the printed page, each table comes in a near term and long term part (figure 5). Each technology item is designated white, yellow,

or red depending on the state of known technology. As illustrated (figure 5), assembly and packaging are areas of concern in 1999.

Table 59b Assembly & Packaging Technology Requirements—Long Term (continued)

YEAR TECHNOLOGY NODE	2008 70 nm	2011 50 nm	2014 35 nm
Performance: On-Chip (MHz) [E]			
Low cost	840	1044	1250
Hand-held	840	1044	1250
Cost-performance	1400	1800	2200
High-performance	2500	3000	3600
Harsh	100	100	100
Memory (D/SRAM)	175/700	200/900	225/1100
Performance: Chip-to-Board for Peripheral Buses (MHz)			
Low cost	125	125	150
Hand-held	125	125	150
Cost-performance [F]	175/700	200/900	225/1100
High-performance [G]	1250	1500	1800
Harsh	100	100	125
Memory (D/SRAM) [F]	175/700	200/900	225/1100

Solutions Exist *Solutions Being Pursued* *No Known Solutions*

Figure 5: Example Chart from ITRS

The Petaflops Planner

Abstract Supercomputer Representation

The planner represents a supercomputer as a collection of SMP nodes. Each node is comprised of chips defined by Silicon area and pin count. The wires connect the pins. The planner then calculates system performance assuming the system is assembled according to on of the architectures described above. The planner assumes each chip, pin, and wire will perform to its maximum capacity. System cost is computed by using ITRS cost figures for ASICs, MPUs, DRAMS, etc.

The illustration of a PIM (figure 6) illustrates the planner's level of abstraction. The rectangle is displayed of a size proportional to the chip area. The yellow area represents RAM on a PIM and blue and red represent processor cores, with the amount of each color proportional to the Silicon area consumed by the component. The planner deals only with area, not specific circuitry.

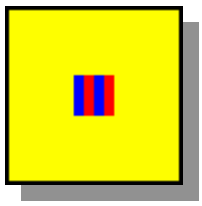

Figure 6: PIM Chip Layout Model

Interconnect

The Petaflops Planner assumes a 3-d mesh topology. The rationale is as follows:

- By choosing a 3-d mesh topology, we trivialize analysis of the mapping between the supercomputer topology and the topology of the machine room. All machine rooms in the universe are 3-d, because the universe is 3-d (spatially). Thus, we can put a 3-d supercomputer into a machine room without having any long wires. The “fat tree” is other popular interconnect topology. Putting a “fat tree” supercomputer into a 3-d machine room involves a spatial mapping and some long wires. Long wires introduce both cost and delay.

- Sandia National Laboratories (SNL) has a history of procuring 3-d mesh supercomputers that remain the “fastest computer on earth” longer than anybody expects. Hence, there is no evidence of a performance penalty by limiting the analysis to 3-d meshes.

Derated Flops

The planner uses a method we call “derated flops” to calculate system performance. The rationale is as follows: Imagine that system performance was based solely on peak FLOPS, and that we would use an architectural optimization method (as we do). With peak FLOPS as the objective function, an optimizer would be expected to find systems where the chips were packed solid with floating point units. These systems would not perform well on real applications because they would lack memory and IPC IO. We found that by “derating” the flop rate so that balance was preserved, we got an objective function suitable for optimization.

We define the “derated flops” for a system as the maximum number of FLOPS that can be claimed while still meeting the balance factors. For example, say the balance factors call for one byte of memory bandwidth per flop. A microprocessor with 10 GFLOPS floating performance and 1 gbyte/second memory bandwidth would therefore be unbalanced. However, we permit the vendor to claim a “derated flops” rate of 1 GFLOP for the microprocessor. One GFLOP is the

highest flop rate that could be claimed without violating the balance factor for memory bandwidth.

Other Factors

We considered two other factors in designing the planner:

Technology derating factor. Observers have long noticed that industry puts the best technology into consumer and commercial products where volumes and profits are highest. Supercomputing tends to get technology a couple years later. To capture this behavior, the planner lets the user specify a “technology derating factor.” Certain ITRS projections get multiplied by this factor to model this delay. The figures that get cut include ASIC density and clock rate, but do not include commodity microprocessor performance and memory density (because supercomputers can include the latest commodity components).

Power. The planner calculates power consumption, but does not use it as a constraint. The calculation applies CMOS power scaling rules to typical values for today’s microprocessors and memories. (It would be desirable for the planner to know maximum power dissipation per chip and then stop allocating components when that power has been reached.)

Architectural Optimization

The planner can optimize designs. The architectural alternatives being considered are parameterized by one or two integers representing the number of CPU cores or memory chips. The planner picks the optimal value(s) for these parameters by iterating over all reasonable values of these integers and noting the value producing the lowest cost per derated flops ratio.

Implementation

We constructed the Petaflops Planner to implement the design process described above. The planner was written in C++ for Windows and Linux and acts as an Internet Web server. That is, the planner starts and listens on a TCP/IP port for Web page accesses from a browser. A user accessing the planner via a browser sees a small Web site with help files, forms for specifying balance factors, and dynamic pages representing candidate supercomputers.

The diagram (figure 7) shows the Web form where the user specifies balance factors.

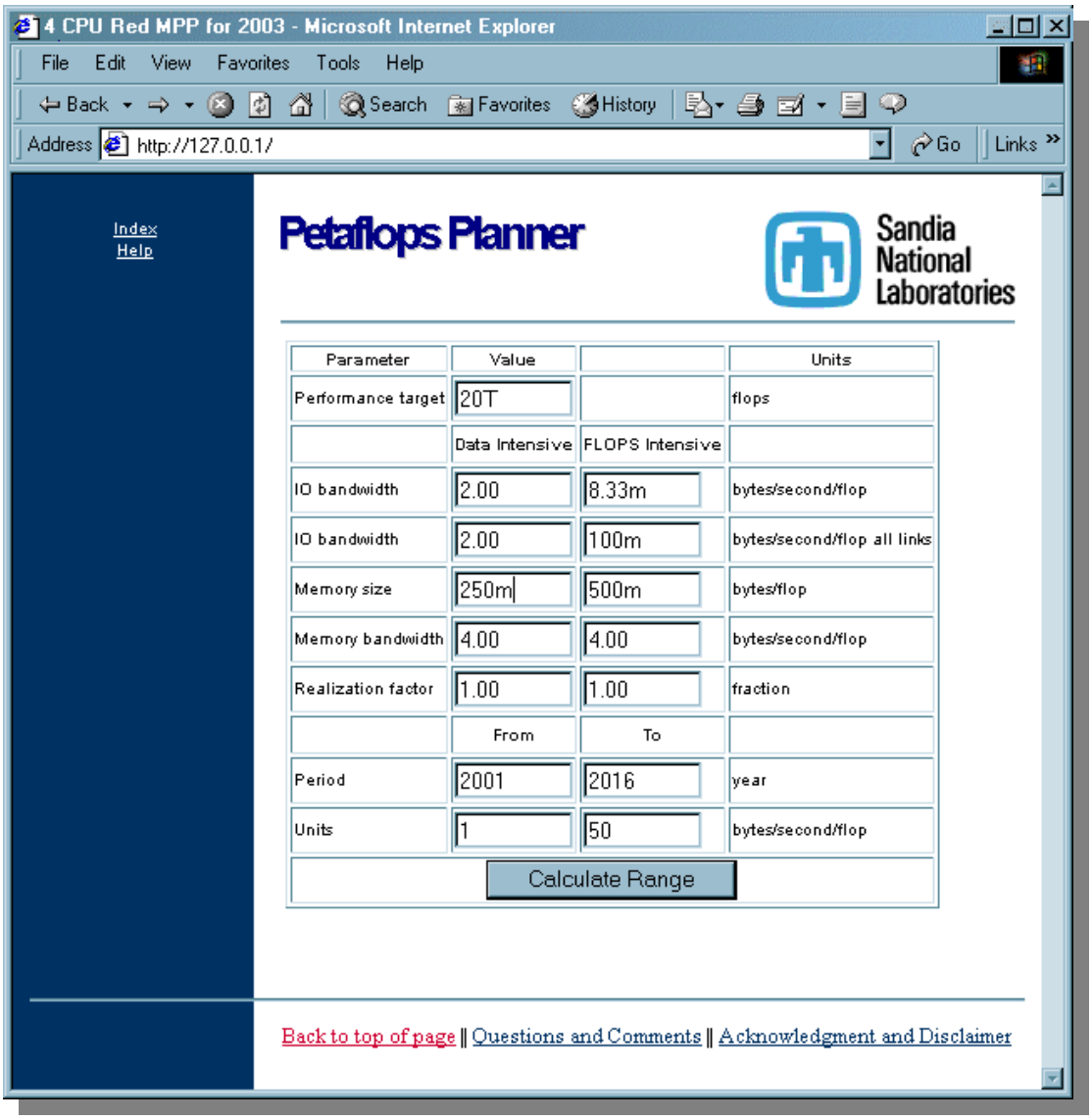

Figure 7: Petaflops Planner Input Form

Upon receiving balance factors, the planner computes several dozen optimal designs and provides a table of hyperlinks for details on each design. The table (not shown) has rows corresponding to year within the range of the ITRS projections and columns corresponding to four candidate architectures and packaging styles. The values in the table cells report the optimal number of CPU cores for the designated architecture in the designated year. Clicking on the hyperlink in the table cell yields a detail page for the design.

The diagram (figure 8) shows a detail page. The planner shows a diagram of a node board, scaling the squares representing chips proportionally to the necessary chip area. The parameters (size, power dissipation, memory capacity)

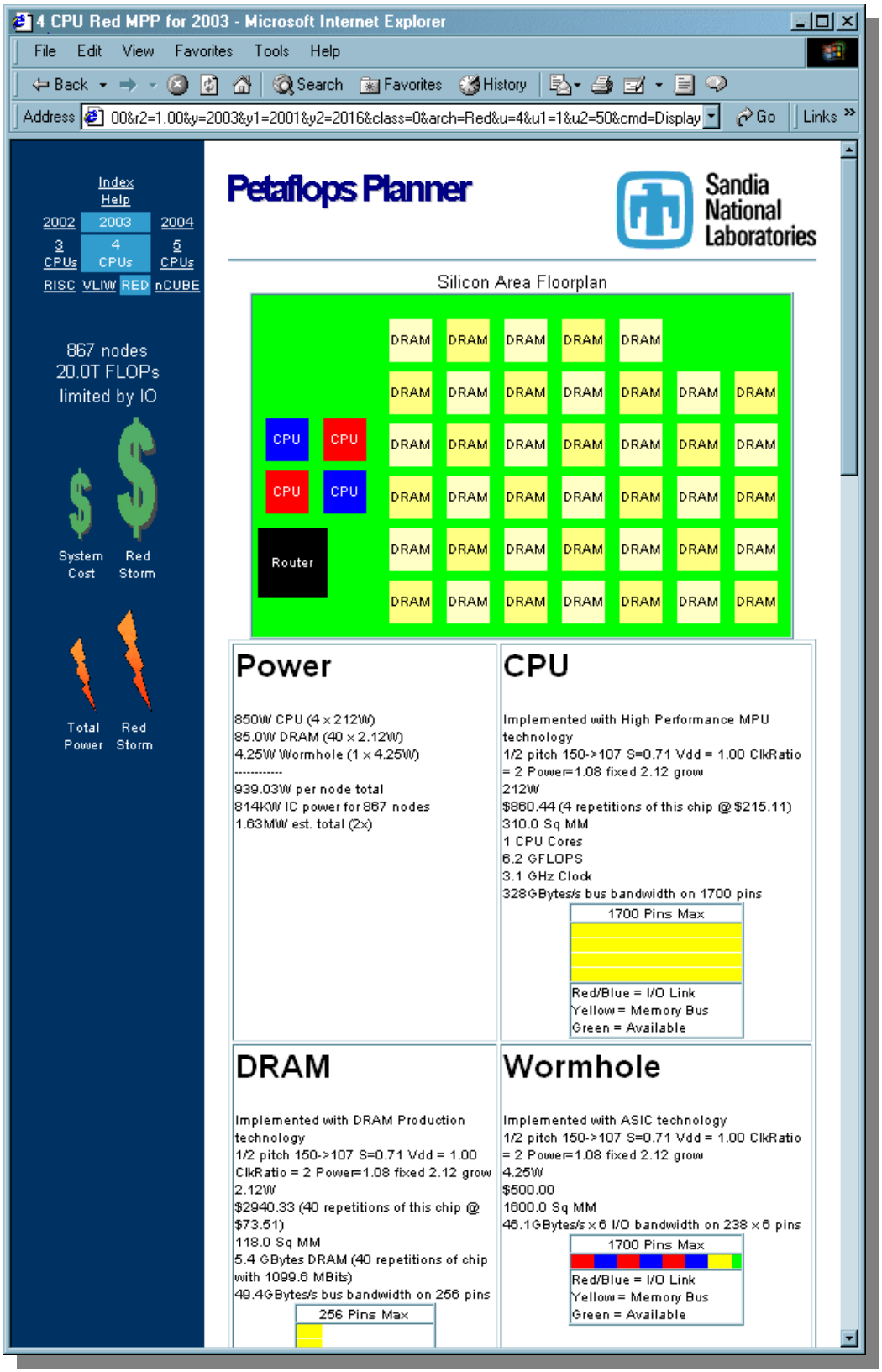

Figure 8: Petaflops Planner Output Page

of each type are reported textually. A bar graph shows the maximum number of pins on a chip and their allocation. The upper-right of the Web page includes a navigation panel that will display detail pages for designs in other years, architectures, and numbers of processors.

Results of Using the Petaflops Planner

The Tool Worked

We have had no problems with stability of the optimization algorithm. Furthermore, we've reviewed the planner's output with various people at Sandia and the results appeared plausible. The tool's speed is acceptable: a 266 MHz PC takes about 2 seconds to perform the ~10,000 design evaluations resulting from submitting balance parameters.

Moore's Law Holds

We wanted to know if supercomputer performance would continue to increase at historical rates to the PFLOPS level. To do this, we calculated and plotted (on a log scale) the estimated cost for an optimal 5 PFLOPS supercomputer for each year into the future and for all the architectures. We expected to see one of two behaviors, both represented in the graph (figure 9):

- Straight lines sloping downward, representing continuation of the exponential progress of Moore's Law.

- Lines that slope downwards for while, followed by a leveling off. This would be the expected result if there were a critical, non-scalable technology component: as other technology components pass it by, the non-scalable component dominates the cost.

As shown in the graph, the planner found both behaviors. The shorter lines (ending in 2014) are based on the 1999 ITRS projections and predict failure of Moore's Law, whereas the longer lines (ending in 2016) are based on 2000 ITRS projections and predict continuation of Moore's Law. This discrepancy is described further below.

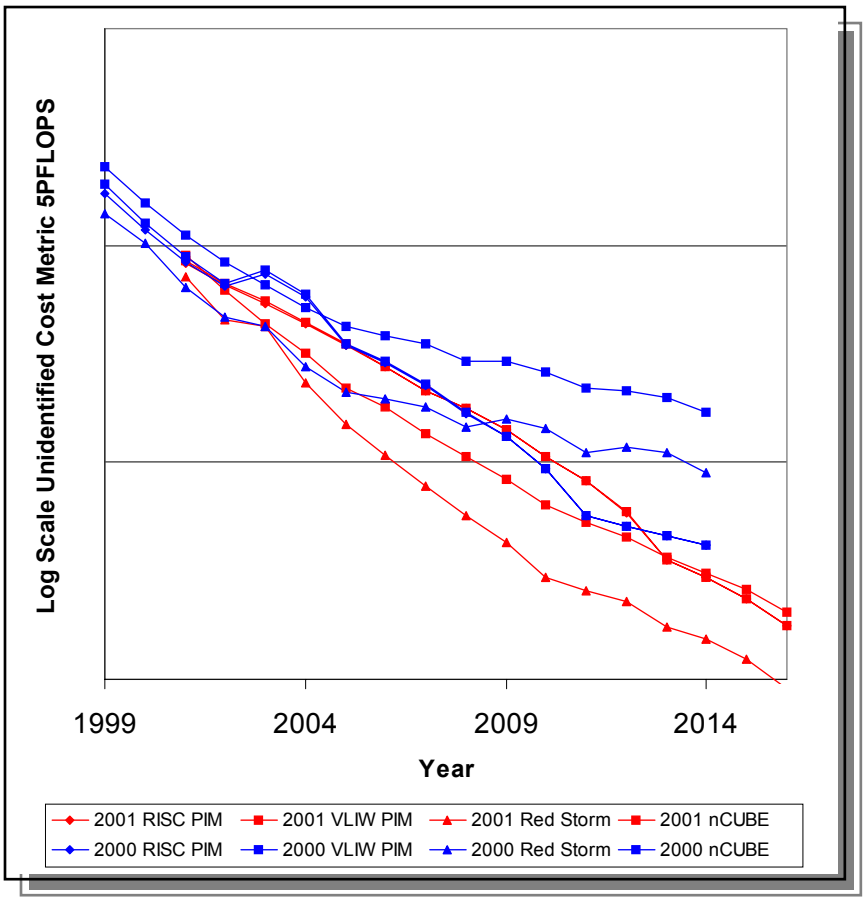

Figure 9: Architecture Costs

Wires and LVDS

The change in conclusion between projections based on 1999 and 2000 ITRS tables illustrates the power and limitations of our approach.

In late 2001 we ran the analysis with the then-current ITRS 1999 tables and were alarmed at the apparent failure of Moore's Law. Examining the optimized designs in the later years revealed enormous chips with thousands of pins with IPC IO busses between chips hundreds of pins wide. Even with such enormous resources applied to chip-to-chip communications, the cores of the chips were "starved" for data. The pertinent ITRS table (figure 5) confirmed the problem: the speed of chip-to-chip communications via pins is limited to several gigabits/second/wire, and the maximum number of pins on a chip is not scaling with Moore's Law. Apparently supercomputers require chip-to-chip bandwidth beyond the scalability of pin-and-wire technology, and as a result Moore's Law fails.

When we ran the planner in early 2002 with the new ITRS 2001 tables, the problem went away. Between 1999 and 2000, a new type of chip-to-chip signaling technology reached the threshold of acceptance and was incorporated into the ITRS 2000 tables. The new technology was Low Voltage Differential Signaling (LVDS, right). Instead of using one wire per bit, LVDS uses two wires driven differentially (+V and -V). The signals are detected by differential amplifiers, where electrical noise cancels. This permits reliable signal recovery at lower power and at higher speed. While LVDS uses twice the pins and wires, the transmission rate becomes scalable with Moore's law. A regular wire is limited to 1 gbit/second whereas a LVDS pair can to 50 gbits/second within the range of projection.

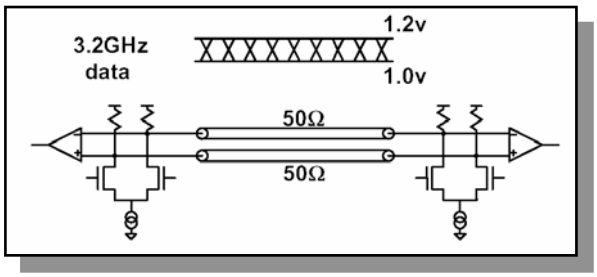

Figure 10: Low Voltage Differential Signaling

No Obvious Winners

No obvious winner emerges from the graph. This is particularly notable because advocates of alternative architectures and packaging styles tout their approaches as being considerably more cost effective than existing systems.

Conclusions

We believe our method has applicability beyond the analysis in this paper. In particular, the ITRS roadmap has adequate predictive power and supercomputer architectures are well enough understood that it is worth the effort to create a roadmap for supercomputer design. Predicting the future is notoriously difficult, as our experience with the change in the ITRS roadmap between 1999 and 2000 illustrated.

We successfully applied optimization techniques to supercomputer architecture. This could be useful in the future.

We conclude that LVDS has a fundamental advantage over pin-and-wire by using a software tool for "scenario analysis" of supercomputer architecture.

There was a widespread belief as recently as 1995, that a petaflop-level computer would be unachievable. This belief is not confirmed.

Future Work

It should be possible to optimize supercomputer for actual applications rather than balance factors. With considerable analysis, one can model the runtime of some “computational kernels” on the abstract hardware like that produced by the planner[ref 4]. The optimizer could then find the architecture that offers the lowest cost per given speed of solution of some application. If this method could be applied to a weighted mix of applications, this would answer the question of “what machine is best for my applications.”

This form of analysis presupposes that computers will be built from the commercial chips covered by the ITRS. It does not account for wafer scale integration or non-CMOS devices. While CMOS technology has a huge maturity advantage over these other technologies, a subsequent analysis should cover them.

References

1. Statement of Work for the Sandia Red Storm Computing Capability Development Program Procurement, RFQ Number 2999, Sandia National Laboratories

2. ASCI Purple RFP, Lawrence Livermore National Laboratory, http://www.llnl.gov/ASCI/purple/

3. 1999 and 2000 International Technology Roadmap for Semiconductors (ITRS), Semiconductor Industries Association (SIA) http://public.itrs.net

4. Predictive Performance and Scalability Modeling of a Large-Scale Application, Darren J. Kerbyson, Henry J. Alme, Adolfy Hoisie, Fabrizio Petrini, Harvey J. Wasserman, Michael L. Gittings, Proceedings of Supercomputing 2001.

Appendix

PIM Design Equations

Optimizer inputs: A node is an SMP with n CPU cores.

The chip area from the ITRS is allocated first to n CPU cores. The remaining chip area is divided by the memory density from the ITRS to get the amount of memory per PIM chip.

The signal pins are divided into 6 bundles, all for IPC IO.

DERATED FLOPS IS MINIMUM OF:	EXPLANATION:
$n \times$ peak FLOPS per core	CPU FLOPS cannot exceed raw capability of CPU cores
$\frac{\text{ASIC I/O bandwidth}}{\text{IPC IO balance factor}}$	Derate CPU FLOPS as necessary to balance IPC IO bandwidth
$\frac{\text{memory capacity per core}}{\text{memory capacity balance factor}}$	Derate CPU FLOPS as necessary to balance memory size

Table 2: PIM Design Equations

Power computed by assuming the chip is x% logic and (100-x)% DRAM. The logic and DRAM scale from current levels by CMOS scaling rules.

NCUBE Design Equations

Optimizer inputs: A node is an SMP with n CPU cores in the ASIC and m DRAM chips; the optimizer picks n and m.

The ASIC's signal pins are divided into 7 bundles: 6 of size $\times$ for IPC IO and one of size y for the memory bus. y = x $\times$ memory bandwidth balance factor / IPC IO balance factor.

Derated Flops is minimum of:	Explanation:
n × peak FLOPS per core	CPU FLOPS cannot exceed raw capability of CPU cores
router chip I/O bandwidth / IPC IO balance factor	Derate CPU FLOPS as necessary to balance IPC IO bandwidth
m × memory capacity per DRAM / memory capacity balance factor	Derate CPU FLOPS as necessary to balance memory size
ASIC bus bandwidth / memory bandwidth balance factor	Derate CPU FLOPS as necessary to balance memory bus bandwidth at CPU
m × memory bus bandwidth / memory bandwidth balance factor	Derate CPU FLOPS as necessary to balance memory bus bandwidth at memories

Table 3: nCUBE Design Equations

ASIC, DRAM and MPU cost from ITRS

Discrete MPP Design Equations

Optimizer inputs: A node is an SMP with n CPU chips and m DRAM chips; the optimizer picks n and m.

Derated Flops is minimum of:	Explanation:
n × peak FLOPS per CPU chip	CPU FLOPS cannot exceed raw capability of CPU cores
router chip I/O bandwidth / IPC IO balance factor	Derate CPU FLOPS as necessary to balance IPC IO bandwidth
m × memory capacity per DRAM / memory capacity balance factor	Derate CPU FLOPS as necessary to balance memory size
CPU bus bandwidth / memory bandwidth balance factor	Derate CPU FLOPS as necessary to balance memory bus bandwidth at CPU
m × memory bus bandwidth / memory bandwidth balance factor	Derate CPU FLOPS as necessary to balance memory bus bandwidth at memories

Table 4: Discrete MPP Design Equations

ASIC $500, DRAM and MPU cost from ITRS

Distribution:

1 MS	9037	J. C. Berry, 8945
1	9019	S. C. Carpenter, 8945
1	9012	J. A. Friesen, 8963
1	9012	S. C. Gray, 8949
1	9011	B. V. Hess, 8941
1	9915	M. L. Koszykowski, 8961
1	9019	B. A. Maxwell, 8945
1	9012	P. E. Nielan, 8964
1	9217	S. W. Thomas, 8962
1	0824	A. C. Ratzel, 9110
1	0847	H. S. Morgan, 9120
1	0824	J. L. Moya, 9130
1	0835	J. M. McGlaun, 9140
1	0833	B. J. Hunter, 9103
1	0834	M. R. Prarie, 9112
1	0555	M. S. Garrett, 9122
1	0821	L. A. Gritzo, 9132
1	0835	E. A. Boucheron, 9141
1	0826	S. N. Kempka, 9113
1	0893	J. Pott, 9123
1	1183	M. W. Pilch, 9133
1	0835	K. F. Alvin, 9142
1	0834	J. E. Johannes, 9114
1	0847	J. M. Redmond, 9124
1	1135	S. R. Heffelfinger, 9134
1	0826	J. D. Zepper, 9143
1	0825	B. Hassan, 9115
1	0557	T. J. Baca, 9125
1	0836	E. S. Hertel, Jr., 9116
1	0847	R. A. May, 9126
1	0836	R. O. Griffith, 9117
1	0847	J. Jung, 9127
1	0321	P. R. Graham, 9208
1	0318	J. E. Nelson, 9209
1	0847	S. A. Mitchell, 9211
1	0310	M. D. Rintoul, 9212
1	1110	D. E. Womble, 9214
1	1111	B. A. Hendrickson, 9215
1	0310	R. W. Leland, 9220
1	1110	N. D. Pundit, 9223
1	1110	D. W. Doerfler, 9224
1	0847	T. D. Blacker, 9226
1	0822	P. Heermann, 9227
1 MS	0818	P. Yarrington, 9230
1	0819	R. M. Summers, 9231
1	0820	P. F. Chavez, 9232
1	0316	S. S. Dosanjh, 9233
1	0316	J. B. Aidun, 9235
1	0813	R. M. Cahoon, 9311
1	0801	F. W. Mason, 9320
1	0806	C. Jones, 9322
1	0822	C. Pavlakos, 9326
1	0807	J. P. Noe, 9328
1	0805	W.D. Swartz, 9329
1	0812	M. R. Sjulin, 9330
1	0813	A. Maese, 9333
1	0812	M. J. Benson, 9334
1	0809	G. E. Connor, 9335
1	0806	L. Stans, 9336
1	1110	R. B. Brightwell, 9224
1	1110	R. E. Riesen, 9223
1	1110	K. D. Underwood, 9223
1	1110	E. P. DeBenedictis, 9223
1	0321	W. Camp, 9200
1	0841	T. Bickel, 9100
1	9003	K. Washington, 8900
1	0801	A. Hale, 9300
1	0139	M. Vahle, 9900
1	9018	Central Technical Files, 8945-1
2	0899	Technical Library, 9616